Tackling the exposed node problem in IEEE 802.11 MAC

M. Tech. Thesis

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology

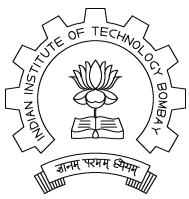
by

Deepanshu Shukla

Roll No: 01329004

under the guidance of

Dr. Sridhar Iyer



School of Information Technology
Indian Institute of Technology, Bombay
Mumbai

Acknowledgment

I express my sincere gratitude toward my guide **Prof. Sridhar Iyer** for his constant help, encouragement and inspiration throughout the project work. Without his invaluable guidance, this work would never have been a successful one. I would also like to thank the members of the *Mobile Computing Research Group* at KReSIT, namely Dr. Leena Chandran Wadia, Srinath Perur, Vijay Rajsinghania, Vikram Jamwal, Anupam Goyal, Satyajit Rai and Abhishek Goliya for their valuable suggestions and helpful discussions.

Deepanshu Shukla
IIT Bombay.
January 19, 2003

Abstract

Ad hoc wireless networks promise convenient infrastructure-free communication. An efficient MAC protocol through which mobile stations can share a common broadcast channel is essential in an ad-hoc network because the medium or channel is a scarce resource. The basic medium access mechanism is basically a Carrier Sense Multiple Access with Collision Avoidance mechanism (CSMA/CA). The CSMA/CA protocol is designed to reduce the collision probability between multiple stations accessing a medium. 802.11 MAC forestalls the possibility of feasible parallel communication by two neighboring nodes that are either both senders or both receivers giving rise to *"Exposed Node"* problem. Exposed Nodes are the nodes which are forced to defer the transmission of data as it is already exposed to an ongoing tranmission. This work adds enhancements to IEEE 802.11 MAC which enables it to schedule concurrent transmissions, thereby improving the channel utilization and solving the Exposed Node problem.

Contents

Acknowledgments **i**

Abstract **ii**

List of Figures **v**

1 Introduction **1**

1.1 Working of multi-hop ad-hoc networks 1

1.2 Brief problem description 2

1.3 Thesis outline 3

2 IEEE 802.11 **4**

2.1 General description of 802.11 architecture 4

2.1.1 How wireless LAN systems are different 5

2.1.2 Destination address does not equal destination location 5

2.1.3 The media impact on the design 6

2.2 Physical Layer characteristics 7

2.3 Components of the IEEE 802.11 architecture 7

2.3.1 The independent BSS as an ad hoc network 8

2.3.2 Distribution system concepts 8

2.4 MAC architecture 9

2.4.1 The Basic access method: CSMA/CA 9

2.4.2 Distributed Coordination Function (DCF) 11

2.4.3 Point Coordination Function (PCF) 12

2.5 Inter Frame Spaces (IFS) and Frame Types 13

2.6 Brief problem discussion 14

3 Problem Identification **16**

3.1 Performance analysis of 802.11 16

3.2 Exposed Node problem 17

3.3 Brief idea of solution 19

4 Detailed solution design **20**
4.1 Design of the exposed node algorithm 20
4.2 Limitations of the algorithm 27

5 Implementation Details **28**
5.1 Glomosim 28
5.2 Architecture of Glomosim 29
5.3 Pseudo Code of exposed Node Algorithm 31

6 Simulation Results **35**
6.1 Effect of physical layer modeling 35
6.2 Network topologies 36
6.3 Results of simulation 37

7 Conclusion & Future Work **38**
7.1 Current proposals and simulated implementations 38
7.2 Proposed changes to 802.11 frames 39
7.3 Related work 39
7.3.1 MACA-P 39
7.4 Future area of investigation 40

Appendices **40**

A Data Structures Used **41**
A.1 Specific to this proposed solution 41
A.2 Node 42
A.3 MAC Layer 43
A.4 Control Packets 44
A.5 Methods specific to this solution 45

Bibliography **46**

List of Figures

1.1 Multihop ad hoc Networks 1

2.1 Distributed System and AP 4
2.2 MAC Interference 7
2.3 Complete IEEE 802.11 architecture 8
2.4 MAC Architecture - DCF and PCF 11
2.5 Transmission of an MPDU using RTS/CTS 12
2.6 IFS Relationship 13

3.1 String Topology 16
3.2 Exposed Nodes 17
3.3 Simultaneous Transmissions in 802.11 18

4.1 Identify Exposed Node 21
4.2 Algorithm Design 25
4.3 Algorithm Design (contd.) 26
4.4 Reverse Exposed Node 27

5.1 Glomosim Architecture 29

6.1 Sample Simulation Topology 36
6.2 Simulation Results - 4 & 6 Node 37

7.1 Standard 802.11 Frames 39
7.2 MACA-P 40

Chapter 1

Introduction

1.1 Working of multi-hop ad-hoc networks

Mobile Ad Hoc Networking technology, long recognized as important in the military communications arena, is receiving increased attention for commercial applications as well. Originally developed under the name Mobile Packet Radio, the technology most generally refers to mobile, multihop wireless networking systems that may operate without the aid of a fixed networking infrastructure. More specifically, Mobile Ad Hoc Networks (MANET) consist of mobile nodes each node consisting logically of a router and one or more attached hosts which communicate using one or more wireless technologies. Such a network is self-organizing in the sense that nodes can join and leave at will with or without requiring preestablished associations.

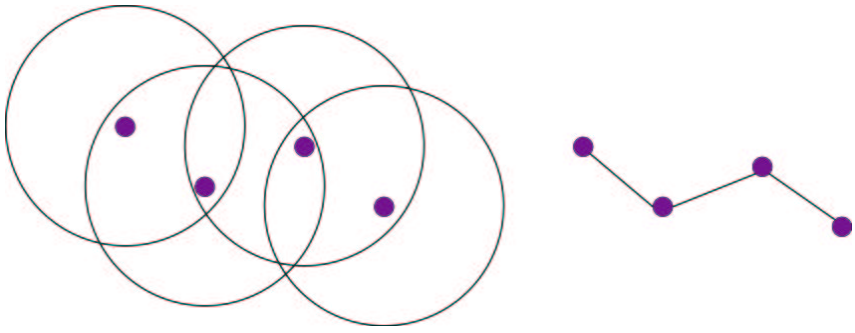


Figure 1.1: Multihop ad hoc Networks

The medium access control (MAC) protocol with which packet-radios (or stations) can share a common broadcast channel is essential in a packet-radio network. CSMA (carrier sense multiple access) [1] protocols have been used in a number of packet-radio networks in the past; these protocols attempt to prevent a station from transmitting simultaneously

with other stations within its transmitting range by requiring each station to listen to the channel before transmitting.

An efficient MAC protocol through which mobile stations can share a common broadcast channel is essential in an ad-hoc network because the medium or channel is a scarce resource. Due to the limited transmission range of the mobile stations, multiple transmitters within the range of the same receiver may not know one another's transmissions, and in effect remain "hidden" from one another. When these transmitters transmit at around the same time, they do not realize that their transmissions collide at the receiver. This is the so called the "Hidden Terminal Problem" [2] which is known to degrade the throughput significantly.

The IEEE 802.11 [3] (hereby referred to as *802.11*) standard is the predominant standard for wireless LAN. Any LAN application, network operating system or protocol, including TCP/IP, will run on 802.11 compliant WLAN as easily as they run over Ethernet. To date the Institute of Electrical and Electronics Engineers (IEEE) have developed three specifications in the Wireless LAN (WLAN) 802.11 family: 802.11, 802.11a, and 802.11b. All three of these specifications use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), as the path-sharing protocol. If a source station has a data packet to send, the station checks the system to see if the path medium is busy. If the medium is not busy, the packet is sent; if the medium is busy, the station waits until the first moment that the medium becomes clear. Testing is done repeatedly by the source via a short test message called Ready to Send (RTS). The data packet is not transmitted until the destination station returns a confirmation message called Clear to Send (CTS). If two stations send at exactly the same time, CSMA/CA prevents the loss of data that might otherwise occur and provides a system for retrying.

1.2 Brief problem description

Several solutions were proposed for the hidden node problem, like MACA[4], FAMA[5] , MACAW [6] to improve the performance and fairness of 802.11 MAC. All these literature highlight the poor performance of the system as a whole because of low utilization of channel[7]. This work deals with improvement of channel utilization by solving the exposed node problem. By default 802.11 CSMA/CA, because of the four way handshake, can not have simultaneous transmissions, which, on the other hand, maybe feasible. This work suggests changes in 802.11 MAC standards to include such transmissions to improve throughput.

1.3 Thesis outline

The thesis defines and explains in detail the working of 802.11 WLAN. In Section 3.2 on page 17 we identify and define the "exposed node" problem in IEEE 802.11 . The proposed solution and the impact of optimization of 802.11 on the existing infrastructure is discussed in Section 7.2. Chapter 4 details out the design which would solve the exposed node problem in 802.11. The next chapter, Chapter 5, details out the actual implementation details and the reason, Section 5.1, for choosing Glomosim [8] as simulation and implementation tool. The simulation results are discussed in Chapter 6. Section 6.1 discusses the impact of physical medium on 802.11 communication and design issues. The related work is discussed in Section 7.3. The thesis concludes in Chapter 7 where we discuss the future work and an overall conclusion of the thesis.

Chapter 2

IEEE 802.11

2.1 General description of 802.11 architecture

The IEEE 802.11 MAC can be used in two modes – *ad-hoc mod*e, or *infrastructure mode*. In ad-hoc mode, a mobile station directly communicates to the another station in transmission range. For the stations which can not communicate directly, a number of stations can co-operate to form a multihop link for communication.

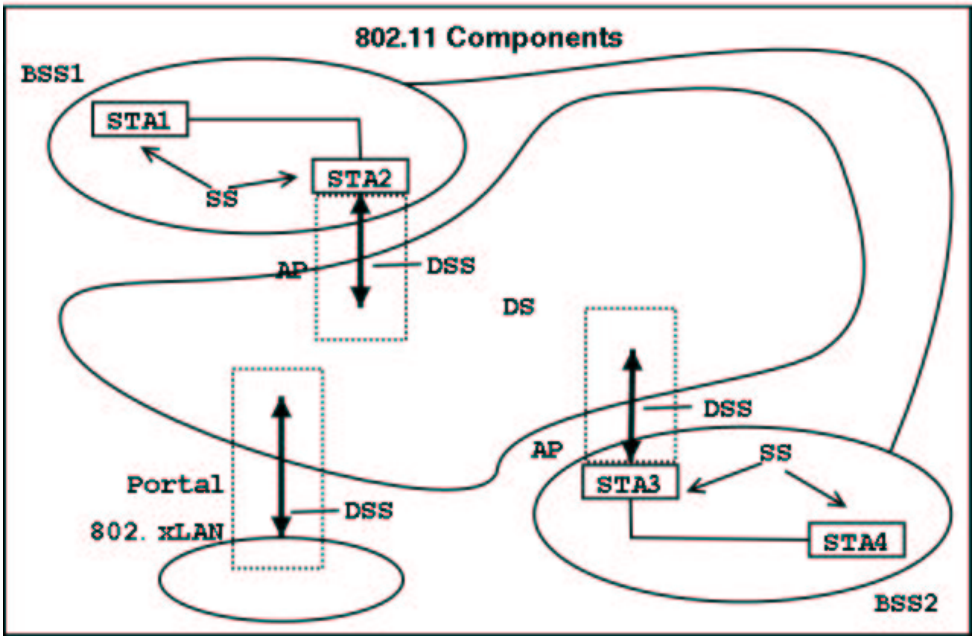


Figure 2.1: Distributed System and AP

In infrastructure mode, one of the stations acts as a base station (or the Access Point (AP)), and all the traffic is coordinated by it. Even if two wireless nodes are in

transmission range of each other, the traffic between those stations goes through the AP. The AP is generally connected to the wired world and a number of APs also may be connected to each other by means of a Distribution System (DS). An AP and its associated mobile stations form a Basic Service Set (BSS).

2.1.1 How wireless LAN systems are different

Two major trends in networking today are support for reservation based applications over wired networks and making connectivity ubiquitous via wireless technologies. There is general belief that networks based on fiber or electrical connections have excellent error characteristic but that wireless networks typically have extremely high error rates . The high error rates in wireless LAN are considered a major challenge. There are some obvious reasons why one would expect wireless connections to have higher error rates than wired connections. Wired connections largely isolate the signal that carries the encoded data from other signals, especially in the case of optical fiber. In contrast, wireless signals share the same propagation medium with many competing signals, and as a result, there are many more opportunities for interference that can result in bit errors. However, wireless connections are very diverse: they differ in range, bandwidth, frequency spectrum used, modulation techniques, interference sources, and physical environment. When signals propagate through space many more factors can influence signal quality than when they propagate in an electrical conductor or fiber. However, characterizing the environment is a critical step in providing a reliable communication service to applications.

2.1.2 Destination address does not equal destination location

Shared channel wireless networks have a unique characteristic that make it very difficult to achieve, or even consistently define, the notion of address mapping. Unlike the wired medium a wireless STA (Station) has access to his own “medium space” and is oblivious to the transmissions going outside that space.

- Spatial (location-dependent) contention for the wireless channel

  Consider a simple channel model where a transmitter has a fixed transmission range, and multiple transmissions in the neighborhood of a receiver will cause a collision at the receiver. Following typical collision avoidance protocols, a successful transmission precludes any station in the neighborhood of either the transmitter or the receiver from engaging in another simultaneous packet transmission/reception. In other words, transmission of a packet involves contention over the joint neighborhoods of the sender and the receiver, and the level of contention for the shared

wireless channel in a geographical region is spatially dependent on the number of contending nodes in the region. This is fundamentally different from wired and cellular channel models, wherein all flows perceive the same contention.

In wired LAN, an address is equivalent to a physical location. This is implicitly assumed in the design of wired LAN. In 802.11, the addressable unit is a station (STA). The STA is a message destination, but not (in general) a fixed location.

2.1.3 The media impact on the design

The physical layers used in 802.11 are fundamentally different from wired media. Thus 802.11 PHY layer in nutshell can be summarized as under

a. Uses a medium that has neither absolute nor readily observable boundaries outside of which stations with conformant PHY transceivers are known to be unable to receive network frames.

b. Are unprotected from outside signals.

c. Communicate over a medium significantly less reliable than wired PHYs.

d. Have dynamic topologies.

e. Lack full connectivity, and therefore the assumption normally made that every STA can hear every other STA is invalid (i.e., STAs may be hidden from each other).

f. Have time-varying and asymmetric propagation properties.

Due to the 802.11 MAC retransmission limits, the consideration of interference and noise significantly increases the data packet drops as the accumulated power of interference signals and noise can increase the probability of frame drops including MAC control frames. As the dropped data packets are not forwarded further to the destinations over multiple hops, the increase in the packet drops at the MAC layer reduces the overall traffic given to the network.

When packet reception event occurs, receiver calculates received signal strength and compares to two thresholds [9]. If the power is below carrier sense threshold, packet is discarded as noise else, if below receive threshold, packet is marked as in error and passed to MAC level. In either of the case PHY medium is reported as busy to MAC. With "carrier sense threshold"[9] approximately half the receive threshold, which makes these values play a significantly important role in MAC performance.

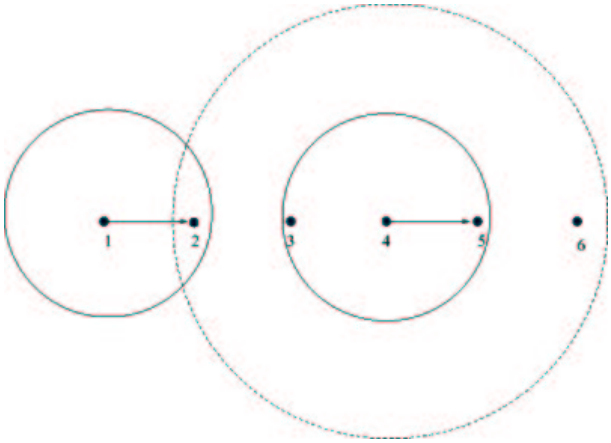


Figure 2.2: MAC Interference

2.2 Physical Layer characteristics

The IEEE 802.11 defines three methods of transmission over the physical layer (PHY): *Infrared*, *Frequency Hopping Spread Spectrum*, and *Direct Sequence Spread Spectrum*. The extended 802.11a uses a physical layer called *Frequency Division Multiplexing* which operates at 5GHz band. The infrared PHY requires the line of sight communication between the transmitter and receiver and hence has limited use. The frequency hopping method transmits and receives over one frequency for a short period of time and jumps to another frequency. The IEEE 802.11 does 1MHz jumps once every tenth of a second. This makes very difficult for an intruder to listen to the transmitter. The transmitter and receiver know the exact hopping pattern in advance. In ISM band 78 frequencies are available and therefore we can have as many as 78 channels theoretically. In the Direct Sequence Spread Spectrum (DSSS) techniquem, the original bit stream is XOR-ed with a chip sequence so as to spread the spectrum of the transmitted signal. With DSSS there are 11 available channels, of which, only three of them are non-overlapping. Therefore we can have only three base stations with overlapping coverage areas.

2.3 Components of the IEEE 802.11 architecture

The basic service set (BSS) is the fundamental building block of the IEEE 802.11 architecture. A BSS is defined as a group of stations that are under the direct control of a single coordination function (i.e., a DCF or PCF). The geographical area covered by the BSS is known as the basic service area (BSA), which is analogous to a cell in a cellular communications network.

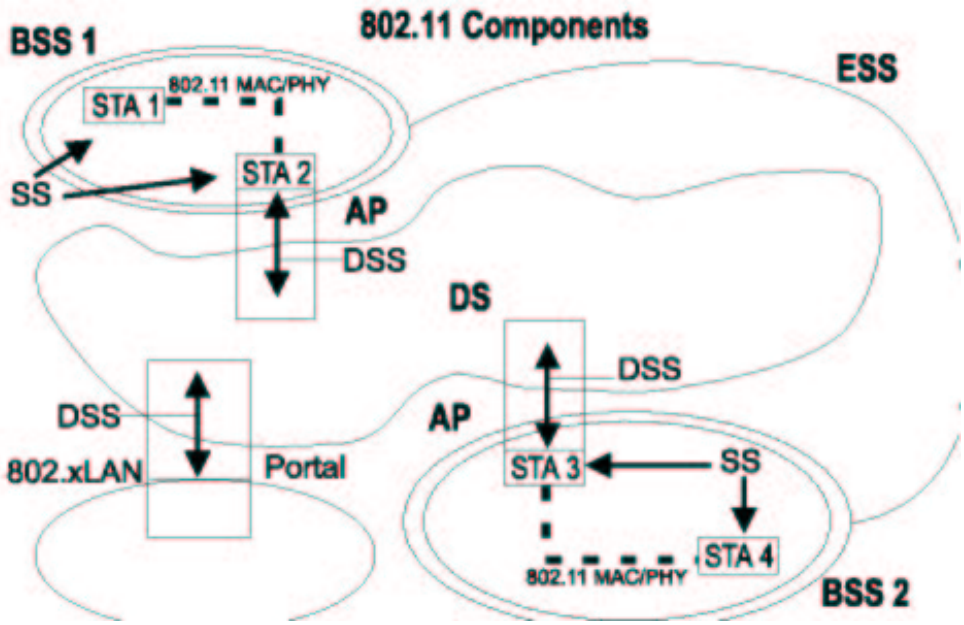


Figure 2.3: Complete IEEE 802.11 architecture

Conceptually, all stations in a BSS can communicate directly with all other stations in a BSS. However, transmission medium degradations due to multipath fading, or interference from nearby BSSs reusing the same physical-layer characteristics (e.g., frequency and spreading code, or hopping pattern), can cause some stations to appear hidden from other stations.

2.3.1 The independent BSS as an ad hoc network

An ad hoc network is a deliberate grouping of stations into a single BSS for the purposes of inter-networked communications without the aid of an infrastructure network. It is the most basic implementation of 802.11 LAN and in the minimum may just consist of two stations. Any node may communicate with its neighbors without the intervention of a centralized access point (AP).

2.3.2 Distribution system concepts

An 802.11 LAN is based on a cellular architecture where the system is subdivided into cells. Each cell (called Basic Service Set, or BSS, in the 802.11 nomenclature) is controlled by a Base Station (called Access Point or, in short,AP). Although a wireless LAN may be formed by a single cell, with a single Access Point, (and as will be described later, it can also work without an Access Point), most installations will be formed by several cells,

where the Access Points are connected through some kind of backbone (called Distribution System or DS). This backbone is typically Ethernet and, in some cases, is wireless itself. The whole interconnected Wireless LAN, including the different cells, their respective Access Points and the Distribution System, is seen as a single 802 network to the upper layers of the OSI model and is knowns in the Standard as Extended Service Set (ESS).

To integrate the IEEE 802.11 architecture with a traditional wired LAN, a final logical architectural component is introduced a portal. Portal is a device that connects between 802.11 and 802 LAN. This concept is an abstract description of the functionality of a "translation bridge".It is possible for one device to offer both the functions of an AP and a portal; this could be the case when a DS is implemented from IEEE 802 LAN components.s

2.4 MAC architecture

The MAC sublayer is responsible for the channel allocation procedures, protocol data unit (PDU) addressing, frame formatting, error checking, and fragmentation and reassembly. The transmission medium can operate in the contention mode exclusively, requiring all stations to contend for access to the channel for each packet transmitted. The medium can also alternate between the contention mode, known as the contention period (CP), and a contention-free period (CFP). During the CFP, medium usage is controlled (or mediated) by the AP, thereby eliminating the need for stations to contend for channel access.

2.4.1 The Basic access method: CSMA/CA

The basic access mechanism, called Distributed Coordination Function, is basically a Carrier Sense Multiple Access with Collision Avoidance mechanism (usually known as CSMA/CA). CSMA protocols are well known in the industry, where the most popular is the Ethernet, which is a CSMA/CD protocol (CD standing for Collision Detection). A CSMA protocol works as follows:

- A station desiring to transmit senses the medium,
- if the medium is busy (i.e. some other station is transmitting) then the station will defer its transmission to a later time,
- if the medium is sensed free then the station is allowed to transmit.

These kind of protocols are very effective when the medium is not heavily loaded, since it allows stations to transmit with minimum delay, but there is always a chance of stations transmitting at the same time (collision), caused by the fact that the stations sensed the medium free and decided to transmit at once. These collision situations must be identified, so the MAC layer can retransmit the packet by itself and not by upper layers, which would cause significant delay. In the Ethernet case this collision is recognized by the transmitting stations which go to a retransmission phase based on an exponential random backoff algorithm.

While these Collision Detection mechanisms are a good idea on a wired LAN, they cannot be used on a Wireless LAN environment, because of two main reasons:

a. Implementing a Collision Detection Mechanism would require the implementation of a Full Duplex radio, capable of transmitting and receiving at once, an approach that would increase the price significantly.

b. On a Wireless environment we cannot assume that all stations hear each other (which is the basic assumption of the Collision Detection scheme), and the fact that a station willing to transmit and senses the medium free, doesn't necessarily mean that the medium is free around the receiver area.

In order to overcome these problems, the 802.11 uses a Collision Avoidance mechanism together with a Positive Acknowledge scheme, as follows: A station willing to transmit senses the medium, if the medium is busy then it defers. If the medium is free for a specified time (called DIFS, Distributed Inter Frame Space, in the standard) then the station is allowed to transmit, the receiving station will check the CRC of the received packet and send an acknowledgment packet (ACK). Receipt of the acknowledgment will indicate the transmitter that no collision occurred. If the sender does not receive the acknowledgment then it will retransmit the fragment until it gets acknowledged or thrown away after a given number of retransmissions. This is how a CSMA system works, we extend the system and use MACA (Medium Access with Collision Avoidance). CSMA/CA works as follows.

When a station wants to send data to another station, it first sends a short Request To Send (RTS) packet to the destination. The receiver responds with a Clear to Send (CTS) packet if its medium is free. On receipt of the CTS, the sender sends its queued data packet(s). If the sender does not receive a CTS after a timeout, it resends its RTS and waits a little longer for a reply. The key to collision avoidance is the effect that RTS and CTS packets have on the other stations of the channel. When a station overhears an RTS addressed to another station, it inhibits its own transmission long enough for the addressed

station to respond with a CTS. Whenever a station overhears CTS addressed to another station it defers its transmission long enough for the DATA to be transmitted. This is the 4-way handshake mechanism, $RTS - CTS - DATA - ACK$ known as CSMA/CA.

2.4.2 Distributed Coordination Function (DCF)

The IEEE 802.11 MAC protocol specifies a Distributed Coordination Function (DCF) which is based on the same RTS / CTS message exchange for unicast data transmissions as the previous MAC protocols. Where 802.11 differs, however, is in its use of collision avoidance before RTS transmission, and its requirement of an acknowledgment (ACK) transmission by the receiver after the successful reception of the data packet. In the case of node mobility, the ACK may also aid in the detection of hidden-terminal interference that was not detectable when the CTS message was sent. This is also defined as "Virtual Carrier Sense" mechanism in the IEEE 802.11 standard specifications.[3]

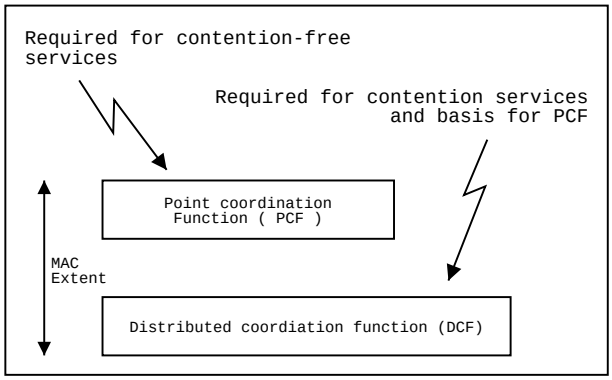


Figure 2.4: MAC Architecture - DCF and PCF

The CSMA/CA protocol is designed to reduce the collision probability between multiple STAs accessing a medium, at the point where collisions would most likely occur. Just after the medium becomes idle following a busy medium (as indicated by the CS function) is when the highest probability of a collision exists. This is because multiple STAs could have been waiting for the medium to become available again. This is the situation that necessitates a random backoff procedure to resolve medium contention conflicts.

All stations receiving either the RTS and/or the CTS, will set their Virtual Carrier Sense indicator (called NAV, for Network Allocation Vector), for the given duration, and will use this information together with the Physical Carrier Sense when sensing the medium.

This mechanism reduces the probability of a collision on the receiver area by a station that is "hidden" from the transmitter, to the short duration of the RTS transmission,

because the station will hear the CTS and "reserve" the medium as busy until the end of the transaction. The duration information on the RTS also protects the transmitter area from collisions during the ACK (by stations that are out of range from the acknowledging station). It should also be noted that because of the fact that the RTS and CTS are short frames, it also reduces the overhead of collisions, since these are recognized faster than it would be recognized if the whole packet was to be transmitted, (this is true if the packet is significantly bigger than the RTS, so the standard allows for short packets to be transmitted without the RTS/CTS transaction, and this is controlled per station by a parameter called RTSThreshold).

The following diagrams show a transaction between two stations A and B, and the NAV setting of their neighbors:

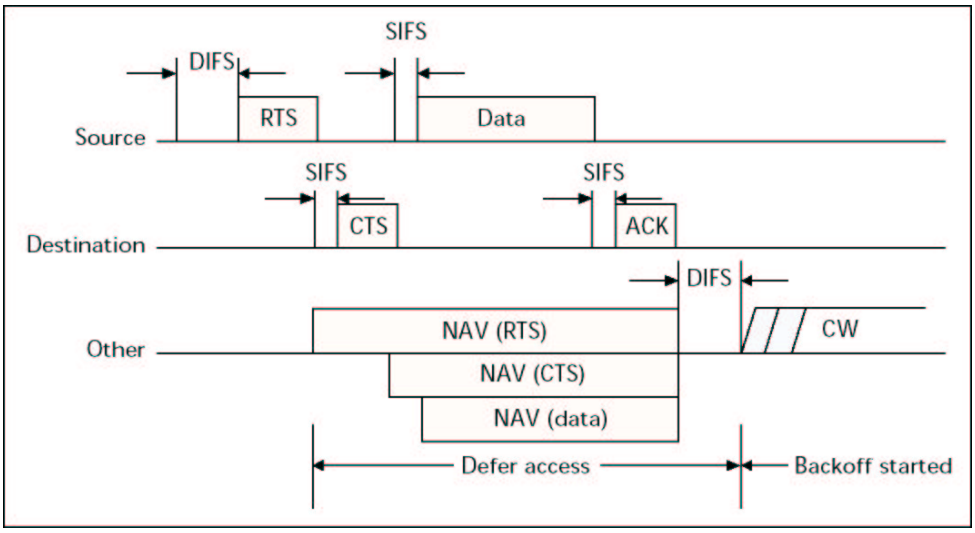


Figure 2.5: Transmission of an MPDU using RTS/CTS

2.4.3 Point Coordination Function (PCF)

Beyond the basic Distributed Coordination Function, there is an optional Point Coordination Function,which which is only usable on infrastructure network configurations and may be used to implement time-bounded services, like voice or video transmission. This Point Coordination Function makes use of the higher priority that the Access Point may gain by the use of a smaller Inter Frame Space (PIFS). By using this higher priority access the Access Point issues polling requests to the stations for data transmission, hence controlling the medium access. In order to allow regular stations the capability to still access the medium, there is a provision that the Access Point must leave enough time for Distributed Access in between the PCF.

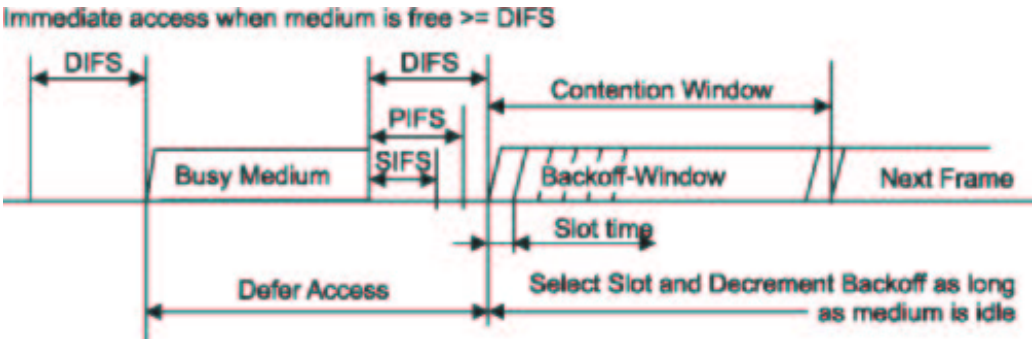


Figure 2.6: IFS Relationship

2.5 Inter Frame Spaces (IFS) and Frame Types

The Standard defines 4 types of Inter Frame Spaces, which are use to provide different priorities:

- SIFS
  Which stands for Short Inter Frame Space, is used to separate transmissions belonging to a single dialog (e.g. Fragment-ACK), and is the minimum Inter Frame Space, and there is always at most one single station to transmit at this given time, hence having priority over all other stations.This value is a fixed value per PHY and is calculated in such a way that the transmitting station will be able to switch back to receive mode and be capable of decoding the incoming packet, on the 802.11 PHY this value is set to 28 microseconds

- PIFS
  Point Coordination IFS is used by the Access Point (or Point Coordinator, as called in this case), to gain access to the medium before any other station. This value is SIFS plus a Slot Time, i.e 78 microseconds.

- DIFS
  Distributed IFS, is the Inter Frame Space used for a station willing to start a new transmission, which is calculated as PIFS plus one slot time, i.e. 128 microseconds

- EIFS
  Extended IFS, which is a longer IFS used by a station that has received a packet that could not understand, this is needed to prevent the station (who could not understand the duration information for the Virtual Carrier Sense) from colliding with a future packet belonging to the current dialog.

The relationship between these timings and frame spacing are shown in Figure 2.6 on page 13

There are three main types of frames:

a. Data Frames : which are used for data transmission

b. Control Frames : which are used to control access to the medium (e.g. RTS, CTS, and ACK)

c. Management Frames : which are frames that are transmitted the same way as data frames to exchange management information, but are not forwarded to upper layers.

Each of these types is as well subdivided into different Subtypes, according to their specific function.

2.6 Brief problem discussion

802.11 MAC forestalls the possibility of feasible parallel communication by two neighboring nodes that are either both senders or both receivers. The PHY layer behavior, as discussed in Section 2.1.3, plays an important role in this anomalous behavior. The key point is that unlike the wired world, "collision" takes place at the recipient and not at the sender. Also the sender would have no knowledge of the collision if it is not in its "sensing range", refer Section 6.1.

The constraints put by the medium become predominantly important as the node switch between Tx [Transmission] and Rx [Reception] mode multiple times in one communication cycle. These constraints by the PHY layer gave rise to the problem of "Hidden Nodes" and "Exposed Nodes". The problem of "Hidden Nodes" was solved by a 4-way handshake MACA, refer [4], protocol for medium access.

The "Hidden Node" is the one which is outside the transmission range of the sender, but within the range of receiver. When a transmission has already begun, this node has no way to find out and senses the medium to be idle nd transmits its own data. However as the node which was receiving the ongoing data is within the range, there will be a collision at the receiver. Hidden terminal is peculiar to wireless (not found in wired). Therefore the node need to sense carrier at receiver, not sender. This leads to a technique called virtual carrier sensing: Sender asks receiver whether it can hear something. If the sender does not reply positively it behaves as if the medium is busy. This is achieved by transmitting RTS (Request To Send), if the recipient node senses its medium idle it responds with CTS (Clear To Send). After which the data is transmitted followed by a positive acknowledgment.

This solved the problem of "Hidden Nodes" but at the same time obviated the possibility of simultaneous transmissions. The "Exposed Node" problem remained unaddressed.

The "Exposed Node" is a node which is in the vicinity of a transmitting station but is not in the vicinity of any receiving stations. Now because of the contagious nature of $RTS - CTS$ mechanism, it extends its NAV and is forced to remain silent for the whole duration of ongoing transmission. The PHY layer constraints enforces that for a successful transmission or reception there can not be a Tx / Rx pair in vicinity. This node however could have transmitted the data safely because it would not have collided with the ongoing transmission, as the collisions take place at the receiver and not the sender. Thus the node, the exposed one, which could have send the data because of the PHY layer characteristics, would now defer its transmission because of MAC protocol.

This work addresses the problem of exposed node and concurrent transmission by utilizing the information heard from the neighboring nodes. It achieves successful overlapping transmission by synchronizing the ACK such that it is concurrent with the ongoing transmission, thus achieving higher medium utilization.

Chapter 3

Problem Identification

3.1 Performance analysis of 802.11

Ad hoc wireless networks promise convenient infrastructure-free communication. We expect the total capacity of such networks to grow with the area they cover, due to spatial re-use of the spectrum: nodes sufficiently far apart can transmit concurrently. However, ad hoc routing requires that nodes cooperate to forward each others packets through the network. This means that the throughput available to each single node's applications is limited not only by the raw channel capacity, but also by the forwarding load imposed by distant nodes. This effect could seriously limit the usefulness of ad hoc routing.

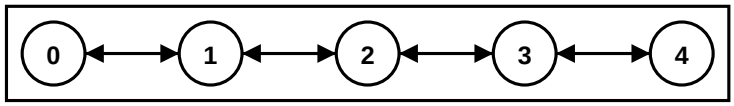


Figure 3.1: String Topology

In a multi-hop network, even relatively small node movement can cause noticeable changes in network topology and thus affect the performance of upper layer protocols, such as throughput and delay. An example of ranking of routing protocols for various scenarios is given in [10]. A high volume of routing queries or updates, caused by mobility or a large number of nodes, causes congestion; the result is not just dropped data packets, but also lost routing information and consequent mis-routing of data.

The problem of Hidden and Exposed node, refer Section 2.6 becomes more acute in a sparse chain type topologies, as shown in Figure 3.1 on page 16, in which case the channel becomes highly underutilized leading to a throughput in the order of few Kbps as compared to expected Mbps. In this work we aim at proposing a solution to the "Exposed Node" problem.

3.2 Exposed Node problem

802.11 uses CSMA/CA to avoid the "Hidden Node" problem. However the four way handshake hinders any simultaneous transmissions such that whenever a node is transmitting, the neighboring nodes will have to remain silent to avoid collision for the full duration of the transmission. This problem is discussed in detail in [11] and also mentions that in the 802.11 MAC there is almost no scheme to deal with this problem.

Unlike the "Hidden Node", an exposed node is the one which is aware of an ongoing transmission but is outside the range of either the sender or the receiver. That is, it hears either $RTS - DATA$ packets or $CTS - ACK$ packets. Such a node hears either of the control packets ie. either RTS or CTS and extends its NAV by the timing specified in these packets.

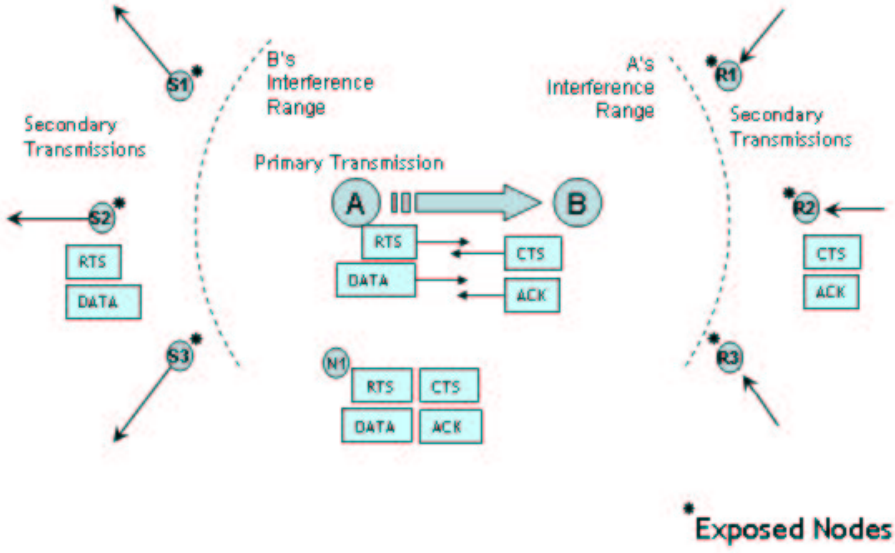
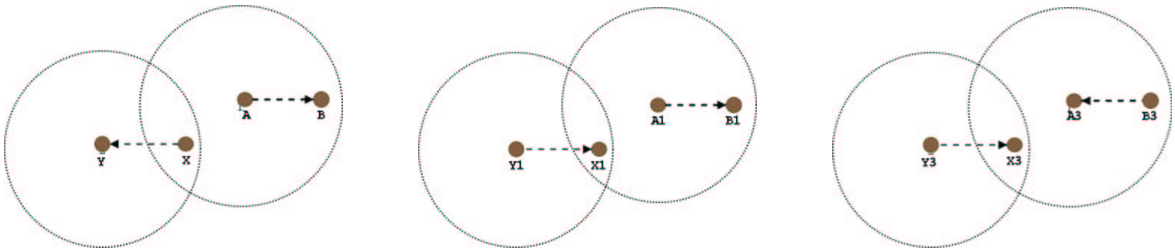


Figure 3.2: Exposed Nodes

The exposed node scenario[1] is shown in Figure 3.2 on page 17. The impact of 4-way handshake on throughput is evident in the figure, as no other node in the vicinity of $A-B$ will start a transmission.

This will lead to lower channel utilization. In the above Nodes $S1$, $S2$ and $S3$ can hear the transmissions of A but not that of B and are outside the interference range of B. The same goes for nodes $R1$,$R2$, and $R3$ as they can listen to B but not A. These nodes are termed as "Exposed Nodes". This is the standard definition of Exposed node, however

[1]The feasible exposed node transmissions are marked as "Secondary Transmissions"



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 3.3: Simultaneous Transmissions in 802.11

we shall use the term Exposed Nodes for $S1 - S2 - S3$, and "$ReverseExposedNodes$" for nodes $R1 - R2 - R3$. These nodes correspond to scenario shown in Figure 3.3 (c). They can hear only CTS and ACK as shown in the figure above.

Lets discuss why 802.11 MAC does not permit two nodes to transmit simultaneously that are either neighbors or have a common neighboring node.

Figure 3.3 on page 18 shows the three possible scenarios in which there can be simultaneous transmissions in wireless transmission. In both the figures placement is such that $X - Y$ and $A - B$ and $X - A$ are in receiving range of each other. Now lets consider Figure 3.3 b, Scenario 2. In this case the node $X1$ would be hearing simultaneous transmissions from $A1$ and $Y1$. This would render both the transmissions useless as there would be a collision at $X1$.

Now consider another scenario, as shown in Figure 3.3 a. In this case if Y is outside the transmission range of A and similarly B is outside the range X, then this transmission would have been feasible had only data to be transmitted. However in 802.11 for transmitting any packet, it follows transmission of $RTS - CTS - DATA - ACK$. In this case the RTS would be transmitted by X or A depending on whose packet arrives earlier. After hearing the RTS one of them ie. A or X would extend its NAV and defer transmission. Same goes for scenario shown in Figure 3.3 c. Upon receipt of RTS the node will transmit a CTS and the node hearing the CTS first would defer its transmission and report the medium as busy.

The reason that 802.11 does not allow any of the concurrent transmission is the inability of the nodes to synchronize transmissions. The nodes do not have the neighborhood information required for such a synchronization. Thus to avoid this, as 802.11 has no mechanism as of now for overlapping data transmissions, the hearing node defers its NAV for any packet not intended for itself.

There are works that identify the low throughput of 802.11 MAC, [10, 11, 2, 12]

because of "Exposed Nodes" and "Hidden Nodes". At the same time they mention that no solution to "Exposed Node" exist in the 802.11 design. In the subsequent sections we propose a solution to the "Exposed Node" problem.

3.3 Brief idea of solution

One of the method that this work proposes is to make the nodes aware of there neighboring transmissions. This would help in synchronization of data in such a fashion that the $DATA$ might be transmitted after the desired interval which would not lead to collision. The concept is :

- As the collision occurred not by transmission of DATA but by the transmission of ACK. So the nodes should synchronize the sending of ACKs so that both (or more) ACKs may be received simultaneously. This would then not result in collisions.
- The above would solve the exposed nodes problem of Nodes $S1 - S2 - S3$ in Figure 3.2 on page 17, by allowing them to have "Secondary Transmissions" which would be interleaved within the Primary "Transmission" of Nodes $A - B$. However the Nodes marked $R1 - R2 - R3$ have no information to synchronize data and correspond to Scenario 3 in Figure 3.3 on page 18. In this work we did not conceive an amicable solution for the "Reverse Exposed Node" scenario.

The above was implemented and the data structures, refer Appendix A, required for maintaining the neighborhood information were added to Glomosim. Timing synchronization details and relevant details are discussed in Chapter 4. Hereby a solution to the exposed node is proposed and implemented in Glomosim, and is discussed in detail in later sections.

Chapter 4

Detailed solution design

The proposed changes enable the node to transmit an overlapping data transmission by synchronizing ACKs with the ongoing transmission. In this solution each node maintains the neighborhood information and based on this information it identifies itself as "exposed" to certain nodes depending on the packets it has listened, refer Figure 3.2 on page 17. The exposed node then squeezes in an overlapping data transmission within the ongoing data transmission. The data is scheduled by the exposed node in such a fashion that the incoming $ACKs$ are concurrent. Thus with each packet of an ongoing transmission, the exposed node successfully sends one of its own packet, thereby increasing channel utilization.

4.1 Design of the exposed node algorithm

The MAC layer at each node responds to the occurrence of the following event:

- Receive a packet from Radio
- Receive a packet from LLC
- Transmit a Packet
- Timer expiry *(waiting for CTS, waiting for DATA, waiting for ACK, waiting for BO: Back Off)*

The idea behind the following algorithm is that whenever a data transmission is going on the node will try to squeeze in a "Secondary Transmission" to get a better overall throughput.

The algorithm can be conceptualized in three parts:

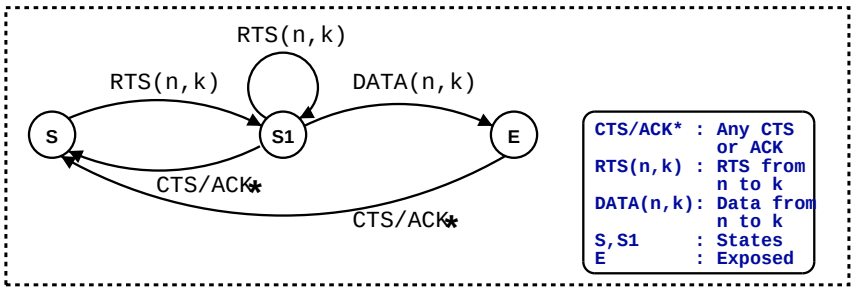


Figure 4.1: Identify Exposed Node

a. Identify exposed node

b. Calculate and synchronize time

c. Transmit $DATA$

A node identifies itself as "Exposed" to another node in its neighborhood if it can hear partial transmissions. If the node is able to hear RTS and $DATA$ then it concludes that it is "Exposed" to the transmitting station. For example in Figure 3.2 on page 17 Nodes -$S1 - S2 - S3$ are exposed to A. Once the node identifies itself as exposed it retains that information unless it either hears an CTS or ACK, either from the same station or any other station. If the node is aware of any "Receiver Node" in the neighborhood then it refrains from indulging in any Secondary Transmissions. The neighboring information is updated by overhearing packets on the medium. This identification takes one transmission cycle, ie it hears the full round of $RTS - CTS - DATA - ACK$. This statistics is maintained for all the nodes in the network. Thus the node can schedule and synchronize data even when it is exposed to multiple nodes.

Once the Node is identified to be exposed, it can participate in Secondary Transmission by "intelligently" scheduling its data. The foremost condition is that the data that has to be transmitted should be smaller in size so that it can be overlapped. This is necessary otherwise the ACK for the "Primary Transmission" will arrive while a secondary data transmission is going on. This will lead to loss of data which is undesirable. To avoid such a situation we keep track of the size of overheard $DATA$ packet in the previous transmissions.

The timing information is calculated for the secondary transmission such that the $ACKs$ from all the transmissions are received at the same time. The timing is calculated from the duration for which the medium is reserved in RTS message.

```
hdr->duration =
      extraPropDelay + SIFS +
      ctsOrAckTransmissionDuration +
      extraPropDelay + SIFS +
      packetTransmissionDuration +
      extraPropDelay + SIFS +
      ctsOrAckTransmissionDuration + extraPropDelay;
```

From this *duration*, we calculate the expected time at which the ACK of the ongoing transmission will be transmitted . The propagation delay is kept into consideration as the node maintains the timing information of the last RTS nd ACK that it overheard. From the difference between the actual packet time and the expected packet time, it calculates the exact time at which the other node will start transmitting ACK. From this time the propagation delay required by its own $DATA$ packet is subtracted and the exact transmission time is calculated. Although the timing is calculated as accurate as up to 1 ns (nanosecond), we have a tolerance of $\frac{1}{2} * DIFS$ for successful reception of all the $ACKs$.

The algorithm defined below predicts the behavior of node and the procedure it follows to initiate overlapping transmissions.

```
Step 0: Initialize MAC and Node
      Set node status "BUSY"
      Initialize all variables
Step 1: MAC idle or waiting for Timer Expiry
Step 2: Which event ?
case a: Received packet from Radio
 if ( my packet? )  then goto Step 3 else goto Step 4
case b: //Receive packet from LLC
    buffer LLC packet
    if (NAV != 0)
         wait for NAV
    else
       if(radio == IDLE)
          Transmit RTS after SIFS period
          set CTS_Timeout Timer
       else
          set BO Timer //BO:: Back Off
case c: //Timer expired
```

```
   What is MAC waiting for ?
   case CTS:
      if(no of tries < RTxThresh) then //RTxThresh:: Retransmit Threshold
         discard packet
      else
         set BO Timer
   case DATA:
      reset MAC state to IDLE
   case ACK:
      if(no of tries < RTxThresh) then
         discard packet
   case BO:
      Tranmit the queued packet after SIFS
      set CTS_Timeout Timer
   end case
end case
goto step 1
```

```
Step 3: //My Packet
   What is the TYPE of packet
   case RTS:
      if (NAV == 0 and status(RADIO) == IDLE) then
         Set MAC state and start CTS_Timeout Timer
         Transmit CTS after SIFS duration
      else
         Discard Packet
   case CTS:
      if (Expecting CTS) then
         Cancel CTS_Timeout Timer
         Transmit DATA after SIFS duration
      else
         Discard
   case ACK:
      if (Expecting ACK) then
         Cancel ACK_Timeout Timer
      else
         Discard
   case DATA:
      if (Expecting DATA) then
         Cancel DATA_Timeout Timer
         Set ACK_Timeout Timer
         Transmit ACK after SIFS
      else
         if ( DATA marked Exposed Pkt)   then
```

```
            Set ACK_Timeout Timer
            Transmit ACK after SIFS
         else
            Discard
   case BX:  //Broadcast
      if(MAC Idle)
         then Process Frame
      else
         Discard
   end case
   goto Step 1
```

```
Step 4: //Not my Packet
   What is the TYPE of packet?.
   case RTS:
      if (node status "EXPOSED")
         if(LLC has packet to send)
            calculate the delay ('t') required
            check for the size of packet.
            if (size < MAXallowed) transmit after 't'
      else
         set Node Status "WAIT"
   case CTS:
      set RESET flag
      set Node Status "BUSY"
   case DATA:
      search for RTS in the recorded messages
      if (found RTS)then
         if (RESET flag set) then
            mark node status BUSY
         else
            mark node status EXPOSED
      else
         discard packet
   case ACK:
      set RESET flag
      set Node Status "BUSY"
   end case
   goto Step 1
```

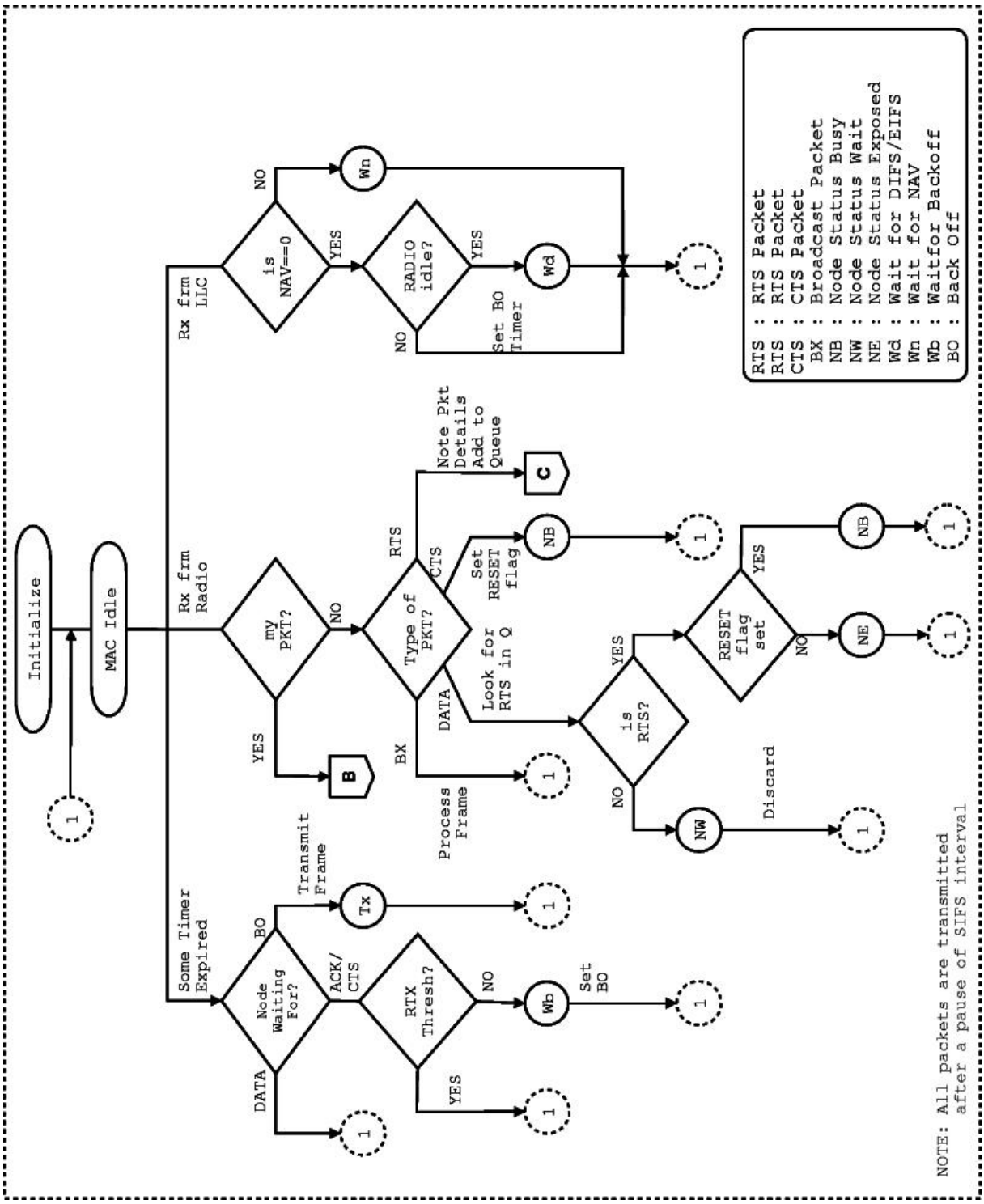


Figure 4.2: Algorithm Design

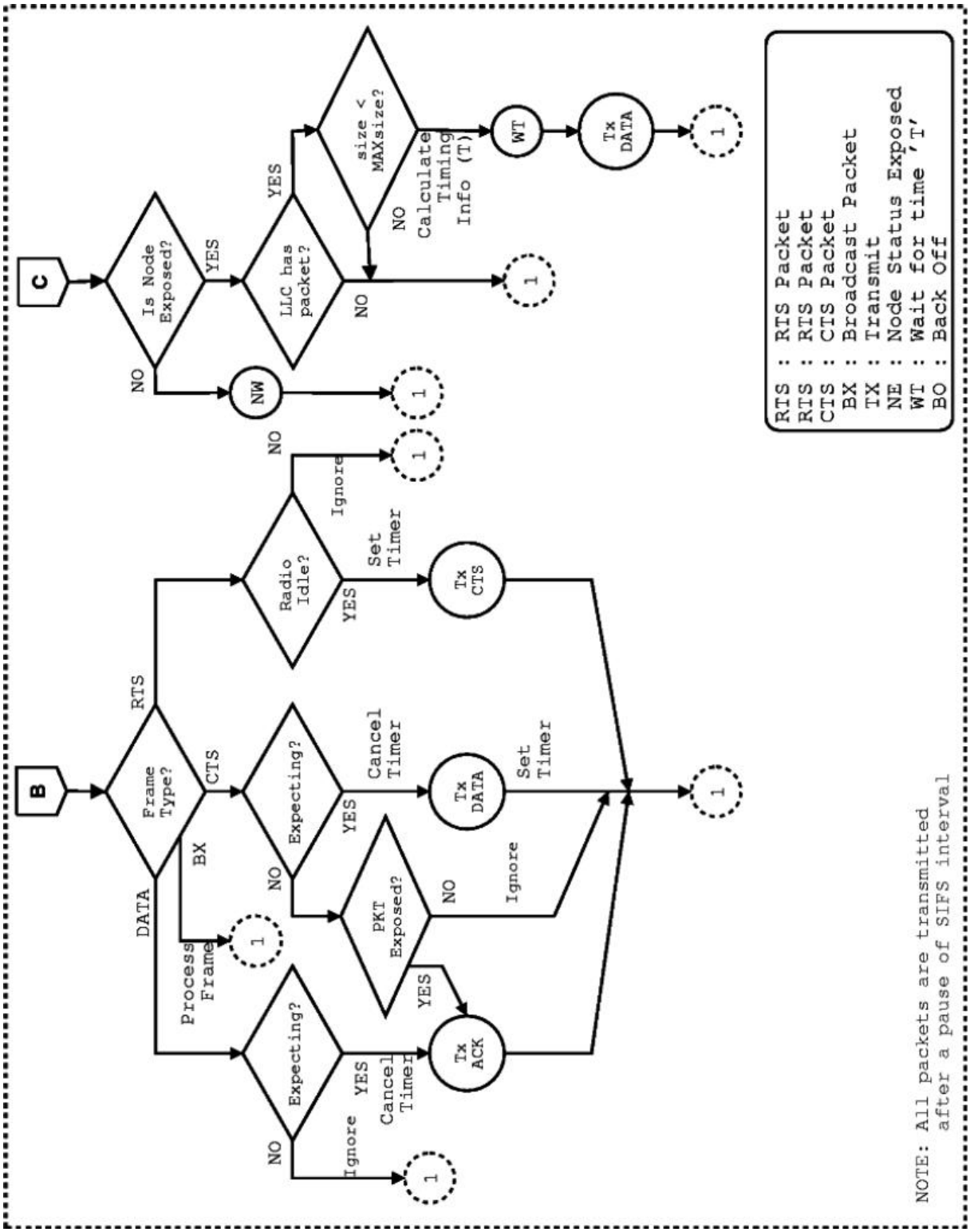


Figure 4.3: Algorithm Design (contd.)

4.2 Limitations of the algorithm

There are a few aspects of this problem which this algorithm does not resolve. The main issue is that of "Reverse Exposed Nodes", denoted by $R1 - R2 - R3$ in Figure 3.2 on page 17. This algorithm deals neither with the identification of these nodes as exposed nodes nor their involvement in secondary transmissions. When an exposed node transmits data "blindly", ie. it just follows a two way handshake, it has full information about the timing and size of data to be transmitted. This however is not true with "Reverse Exposed Nodes", that node being at the receiving end. As shown in the following figure, $A \Rightarrow B$ is the primary transmission and node D is reverse exposed node.

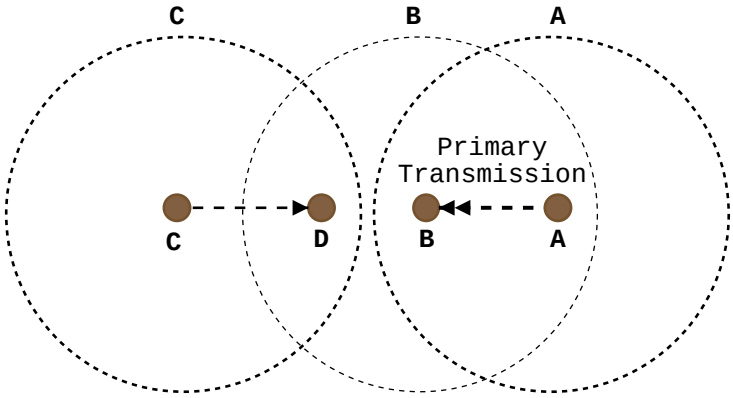


Figure 4.4: Reverse Exposed Node

In this scenario node C has no idea about the timing synchronization nor it has any information about the size of primary data transmission. As in such a case this algorithm can not provide sufficient synchronization to enable the reverse nodes to participate in concurrent transmissions. This problem has been left open, refer Chapter 7.

Chapter 5

Implementation Details

5.1 Glomosim

The proposed changes in the MAC were implemented and verified in Glomosim. Glomosim [8] is a library-based sequential and parallel simulator for wireless networks. It is designed as a set of library modules, each of which simulates a specific wireless communication protocol in the protocol stack. The library has been developed using PARSEC[13], a C-based parallel simulation language. New protocols and modules can be programmed and added to the library using this language. Glomosim has been designed to be extensible and composable. It has been implemented on both shared memory and distributed memory computers and can be executed using a variety of synchronization protocols.

Layer:	Models:
Physical (Radio propagation)	Free space, Rayleigh, Ricean, SIRCIM
Data Link (MAC)	Data Link (MAC)s
Network (Routing)	Flooding, Bellman-Ford, OSPF, DSR, WRP
Transport	TCP, UDP
Application	Telnet, FTP

Table 5.1: Models currently in the Glomosim library

The above table lists the models currently supported in Glomosim. Glomosim is aimed at simulating models that may contain as many as 100,000 nodes with a reasonable execution time. Glomosim has been designed to be extensible and composable: the communication protocol stack for wireless networks is divided into a set of layers, each with its own API. Models of protocols at one layer interact with those at a lower (or higher) layer only via these APIs. The modular implementation enables consistent comparison of multiple protocols at a given layer.

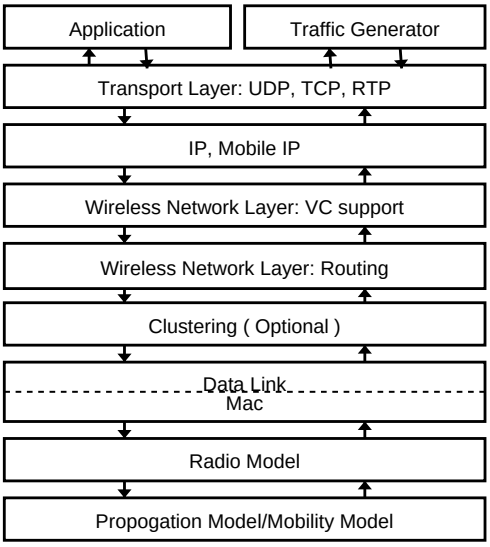


Figure 5.1: Glomosim Architecture

This work led to in-depth analysis of working and coding of Glomosim .This knowledge would help in further development and future implementations and open new horizons for testing and implementing new ideas and protocols in both ns and Glomosim. Glomosim however is COPYRIGHTED software. Release 2.0 of Glomosim is available at no cost to educational users only.

5.2 Architecture of Glomosim

Glomosim aims to develop a modular simulation environment for protocol stacks described in the section Section 5.1, that is capable of scaling up to networks with thousands of heterogeneous nodes. If all protocol models obey the strict APIs defined at each layer, it will be feasible to simply swap protocol models at a certain layer (say evaluate the impact of using CSMA rather than MACA as the media access control protocol) without having to modify the models for the remaining layers in the stack.

Glomosim library is written in PARSEC (for PARallel Simulation Environment for Complex Systems) [13]. PARSEC adopts a message-based approach to discrete-event simulation: physical processes are modeled by simulation objects called *entities*, and events are represented by transmission of time-stamped *messages* among corresponding entities. A number of protocols have been developed at each layer and models of these protocols or layers can be developed at different levels of granularity.

Glomosim assumes that the network is decomposed into a number of partitions and a single entity is defined to simulate a single layer of the complete protocol stack for all the

network nodes that belong to the partition. Interactions among the entities must obey the corresponding APIs.

Simple APIs between every two neighboring models on protocol stacks is predefined to support their composition. These APIs specify parameter exchanges and services between neighboring layers. The simplicity of the APIs allows developers to model their protocols rapidly in an independent fashion. The APIs currently defined in Glomosim are presented:
$ChannelLayerRadioLayerAPIs$:

- DATA PACKET FROM CHANNEL TO RADIO:
  Fields: *payload, packetSize*
  These fields refer to the actual data and size of data being received.
- DATA PACKET FROM RADIO TO CHANNEL:
  Fields: *payload, packetSize*

$RadioLayerMACLayerAPIs$:

- DATA PACKET FROM RADIO TO MAC:
  Fields: *payload, packetSize*
- DATA PACKET FROM MAC TO RADIO:
  Fields: *payload, packetSize*
  Request Channel Status from MAC to Radio: Fields: (none) This message is used by the MAC layer to request information about the current channel status.
- REPORT CHANNEL STATUS FROM RADIO TO MAC:
  Fields: *status, flag*
  This message is used by the radio layer to return the current status of the channel as well as the method by which the information is being reported (passively or actively based on the request message sent by the MAC layer).

$MACLayerNetworkLayerAPIs$:

- DATA PACKET FROM MAC TO NETWORK:
  Fields: *payload, packetSize, sourceId*
  The sourceId refers to the previous hop from which the packet arrived.
- DATA PACKET FROM NETWORK TO MAC:
  Fields: *payload, packetSize, destId*
  The destId refers to the next hop where the packet will travel.

$NetworkLayerTransportLayerAPIs$:

- DATA PACKET FROM TRANSPORT TO NETWORK:
  Fields: *payload, packetSize*

- DATA PACKET FROM NETWORK TO TRANSPORT:
  Fields: *payload, packetSize, sourceId*

UDPTransportLayerApplicationLayerAPIs :
TCPTransportLayerApplicationLayerAPIs

5.3 Pseudo Code of exposed Node Algorithm

Please refer to Appendix A for detailed description of the datastructures used in Glomosim. In this section we shall analyze each of the cases as described in Section 4.1 with respect to there working in Glomosim.

```
ExaminePotentialIncomingMessage (···  )
{  ···
 calculate new Wait;
 node->nodeStatus = checkExposedNode(node, M802, thePacketIfItGetsThrough);
 ···
 if (node->nodeStatus == M802_11_EXPOSED)
 {
  ···
   Mac802_11ForceTransmitFrame( ···  );
 }
 ···
}
```

```
Mac802_11NetworkLayerHasPacketToSend (···  )
{
  if (M802->state == M802_11_S_IDLE)
    {
      if ((!Mac802_11WaitForNAV (node, M802)) &&
          (RadioStatus (node, M802) == RADIO_IDLE))
       {  ···
          assert (M802->BO == 0);
              ⋮
       }
      else
       {
         Otherwise set backoff.
              ⋮
       }
      AttemptToGoIntoWaitForDifsOrEifsState (node, M802);
    }
}
```

```
Mac802_11HandleTimeout(GlomoNode * node,
                       GlomoMac802_11 * M802)
{
···   switch (M802->state)
    {
      case M802_11_S_WF_DIFS_OR_EIFS:  ···
      case M802_11_S_BO: ···
      case M802_11_S_WFDATA: ···
      ⋮
    }
}
```

```
checkExposedNode( GlomoNode * node,
          GlomoMac802_11 * M802,const Message * msg)
{
 //Case of Msg Addresses to itself
 if (hdr->destAddr == node->nodeAddr)return ···
 if (M802->state >= M802_11_S_WFCTS) return ···  /*Waiting for CTS or ACK*/
 if (hdr->destAddr > node->numNodes) return ··· /***** Broadcast */
/* Check for RESET_MSGS */
```

```
  if (hdr->frameType == M802_11_CTS) return ···  /***** CTS */
  if (hdr->frameType == M802_11_ACK) return ···  /***** ACK */
  //Either a DATA or RTS
  //If RTS from an already identified Source, return status as it is,
        update TIME info
 //New Msg ....or DATA msg
 // If there are no Msgs in the queue, then add the msg to the tail
   if (tmpMsg == NULL && hdr->frameType == M802_11_RTS)
     {···
       return M802_11_WAIT;   //return to the calling sub
     }
  switch (hdr->frameType)
     {
      case M802_11_DATA:
        //Check if there exist an RTS from the sender sending the DATA ??
        // If not then simply discard the data packet
            //If yes then most Probably an Exposed node
            //Update neighborhood information
      case M802_11_RTS:
        // Check if its  a retransmit of RTS
        // Add the Packet to the end of list
     }
}
```

```
Mac802_11ReceivePacketFromRadio(···  )
{
if (hdr->destAddr == node->nodeAddr)
  {
   switch (hdr->frameType)
   {
   case M802_11_CTS:
     Mac802_11CancelTimer(node, M802);
     Mac802_11TransmitDataFrame(node, M802);
   case M802_11_ACK:
     Mac802_11CancelTimer(node, M802);
     Mac802_11ProcessAck(node, M802, msg);
   case M802_11_RTS:
     Mac802_11CancelTimer(node, M802);
     Mac802_11TransmitCTSFrame(node, M802, msg);
   case M802_11_DATA:
     Mac802_11ProcessFrame(node, M802, msg);
   }
  }
```

```
}
```

```
Mac802_11ForceTransmitFrame( ··· ) )
{
 QueueWasEmpty = NetworkIpOutputQueueIsEmpty( ··· );
 if (!QueueWasEmpty)//Network HAS packet to send
 {
  NetworkIpOutputQueueTopPacket(··· )
  if (tmpPktPtr != NULL)//Network has Packet to send
  {
   if ( tmpPktPtr->packetSize > packetSize)
     return; //Data packet from Network is larger than allowed
   if (tmpPktPtr->packetSize > M802_11_RTS_THRESH
        && nextHopAddress != ANY_DEST )
    {
     tmpPktPtr->isExposed = TRUE;
     // Calculate timing information
     M802->rtsWaitDuration = packetTime - simclock();
     if (M802->rtsWaitDuration > 0)
      {
       ···
       Mac802_11TransmitDataFrame(node, M802);
      }
    }
  }
 }
```

Chapter 6

Simulation Results

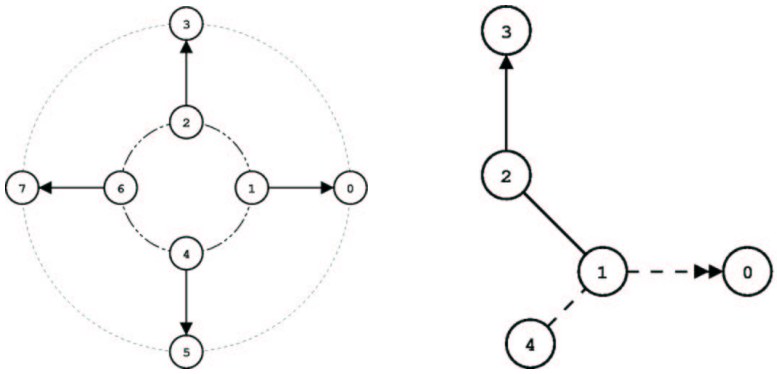
6.1 Effect of physical layer modeling

Due to the 802.11 MAC retransmission limits, the consideration of interference and noise significantly increases the data packet drops as the accumulated power of interference signals and noise can increase the probability of frame drops including MAC control frames. As the dropped data packets are not forwarded further to the destinations over multiple hops, the increase in the packet drops at the MAC layer reduces the overall traffic given to the network.

However, the situation becomes much worse because the radios can interfere with each other beyond the range at which they can communicate successfully. For example, 802.11 nodes in the ns simulator can correctly receive packets from *250 meters* away, but can interfere at *550 meters* and in Glomosim these values are *376 meters* and *627 meters* respectively, under default settings. These values are calculated based on the following physical layer parameters specified in configuration file :

- Pathloss Model ($Free - SpaceModel$, $TWO - RayModel$)
- Radio Type ($Radio - AccNoise$, $Radio - NoNoise$)
- Radio Rx Sensitivity
- Radio-RX-SNR-Threshold
- Radio-RX-Threshold

Hence, in Figure 2.2 on page 7, node 4 s packet transmissions will interfere with RTS packets sent from 1 to 2, preventing 2 from correctly receiving node 1 s RTS transmissions or sending the corresponding CTS. These distances play a major role in the exposed node



(a) For 4 - 6 - 8 Nodes (b) Multihop Topology

Figure 6.1: Sample Simulation Topology

situation as the node would defer its transmission if it senses that it is in vicinity of another receiving node. Further detailed analysis of physical layer impact on the throughput is discussed in [12].

6.2 Network topologies

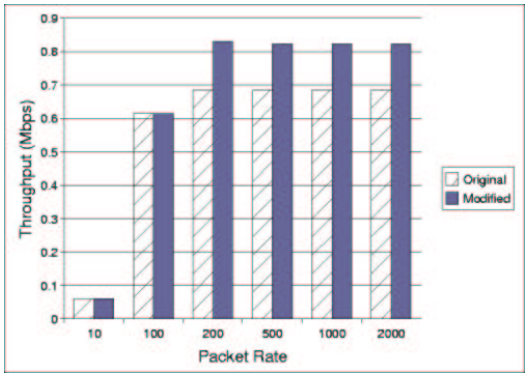
We start with simulating the topology shown in figure 6.1, with data transmission from Node 2 and Node 3 to Node 4 and Node 1 respectively. This topology was further expanded to extend the simulation to 6 and 8 nodes. The Data source was a CBR with data packet size 512 *bytes* and 1024 *bytes*.

The terrain dimension was 1500 $\times$ 1500 *meters*. We have used default Glomosim parameters for Radio Carrier Sense and Radio Receive Threshold. The Rx Receive range is 350 *meters* and interference range is 637 *meters*.

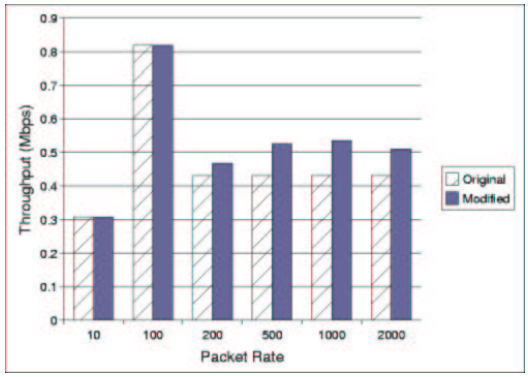
The carrier sense and receive threshold range play a major role in the performance. The ratio of these ranges is 1:1.8. It can be shown mathematically that with these values it is not possible to have more than two concurrent transmissions at any give time. In case of $8 - nodes$, only two nodes at any given time participate in Secondary Transmissions while two are the Primary Transmissions. The scenario for multihop simulation is shown in figure 6.1 (b). Node 4 is sending 1024 *bytes* packets to Node 0, and Node 1 is sending 512 *bytes* packets to Node 3. This makes Node1 as the bottleneck.

6.3 Results of simulation

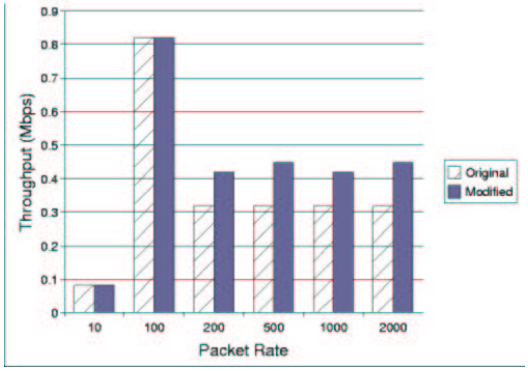
The graphs obtained after simulation showed the relative increase in throughput. It can be noticed that only in the case of higher packet rate this increase is substantial. In low packet rate the packets are already interleaved and there are no packet drops because of medium busy.



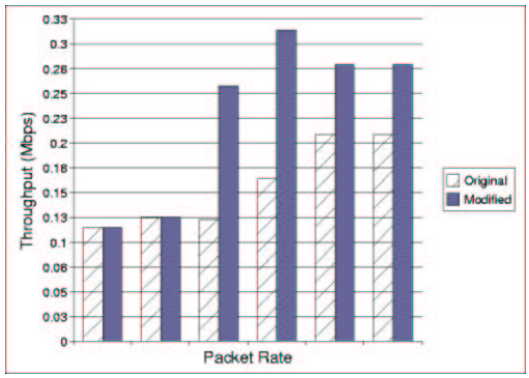
(a) 4 Nodes



(b) 6 - Nodes



(c) 8 - Nodes



(d) 5 - Nodes, Multihop

Figure 6.2: Simulation Results - 4 & 6 Node

There was no performance increase at lower rates as shown above. However at higher rate as the overlapping transmissions are successfully scheduled we get an increase in throughput. At very high rates 802.11 utilization becomes constant and the utilization of the modified 802.11 MAC is more. In multihop scenario with 5 nodes the modified MAC shows substantial increase in throughput with performance as high as 190% in some cases.

Chapter 7

Conclusion & Future Work

7.1 Current proposals and simulated implementations

This work identifies the limitation of 802.11 for supporting concurrent transmissions. This is mainly due to the fact that 802.11 does not have any information about the ongoing transmissions in its neighborhood. The lack of overlapping transmissions was responsible for the "Exposed Node" problem which led to degradation in throughput due to insufficient channel utilization.

We modify the default behavior of 802.11 to solve these issues. The solution we propose identifies an "Exposed Node" from the packets heard over the medium. This is achieved by maintaining information about the past transmission. The node then utilizes this information to squeeze an overlapping $DATA$ packet while another transmission is going on.

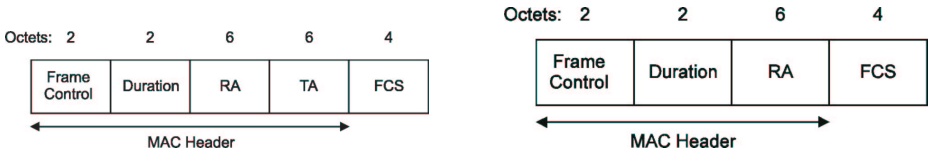
The timing information is calculated from the packets overheard. Based on these timing values the node schedules its transmission such that the ACK from both the transmissions are concurrent. Thus it transmit an extra packet for each time it is exposed to an ongoing transmission. The default behavior of the node remains unaltered. PHY layer plays an important role in successful implementation of this solution as in the wireless world, successful transmission is *not* equivalent to successful reception. This requires more in depth analysis of physical medium and its parameters as discussed in Section 7.2.

The implementation was done in Glomosim. We carried out the simulation for $4-6-8$ nodes distributed as per the topology shown in Figure 6.1 on page 36. The results are discussed in Section 6.3. The simulation achieved 30-35% increase in normal single hop scenarios. It achieved 190% increase in multihop scenarios in one of the cases, while in others it showed sufficient increase in channel utilization.

7.2 Proposed changes to 802.11 frames

This work modifies the control packets by adding an extra field to facilitate the overhearing node in deducing information about the neighbors.

The prime assumption in the design of the control frames being that these frames would be used or processed only by the intended nodes. For this reason we don't need the field TA [Transmitter's Address] in ACK and CTS as these messages are expected. However this information is insufficient to deduce the neighborhood information by a node which is overhearing the conversation. This work adds another field, refer Appendix A.4, "$SRC - Address$" to each of these control packets.



(a) RTS Frame

(b) CTS / ACK Frame

Figure 7.1: Standard 802.11 Frames

Another information that has to be conveyed is about the message itself. In IEEE 802.11 the node drops any $DATA$ packets which it receives unexpectedly, ie. it has not replied with RTS to that particular sender. In our work this behavior has been overridden and an extra $isExposed$ flag is added in the message header. Whenever a node receives an unexpected $DATA$ packet it replies with ACK *only* if the $isExposed$ flag is set, otherwise its ignored. Whenever a data transmission is initiated by an exposed node this flag is set.

7.3 Related work

7.3.1 MACA-P

Arup Acharya and Archan Misra of IBM Research Division, released an IBM Research Report, under "Limited Distribution Notice", which dealt with concurrent transmissions in multi-hop wireless networks. They had given it a nomenclature of MACA-P. MACA-P is a RTS/CTS based MAC protocol that enables simultaneous transmissions in multi-hop adhoc networks. MACA-P is a set of enhancements to the 802.11 MAC that allows parallel transmission in many situations when two neighboring nodes are either both receivers or transmitters. Unlike 802.11 the data transmission is delayed by a control

phase allowing multiple sender-receiver pairs to synchronize there data transfer. They obtained the performance of the system by implementing it in ns.

MACA-P's design seek to increase the feasible set of concurrent transmissions by introducing a control gap between the RTS/CTS and $DATA/ACK$ phase. This allow two neighboring senders to synchronize the start of DATA and ACK. It introduces an *additional* control message RTS''.

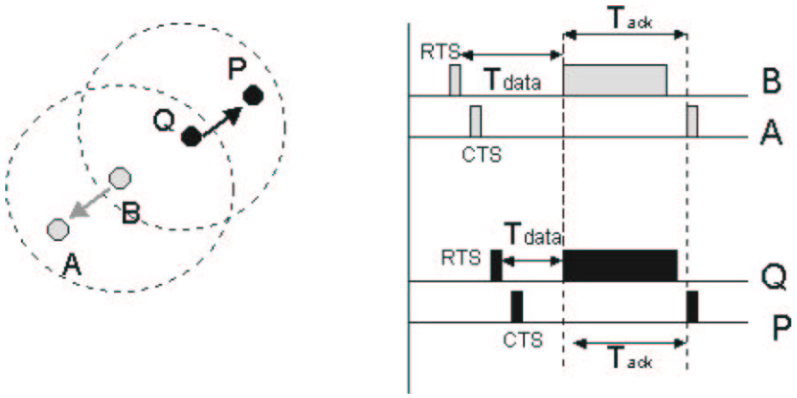


Figure 7.2: MACA-P

It includes the time for T_ACK and T_DATA in the control packets RTS and CTS by adding an extra 2-byte field for each of them. The recipient node which intends transmitting data, carries on $RTS - CTS$ exchange with the desired node within the "Control Phase" period. The actual data by the nodes start at the time specified by T_DATA. MACA-P also discusses the role of interference range and receive range on the performance of the protocol. They acknowledge that because of the PHY layer modeling factors the performance can degrade in dense environment. MACA-P also solves the problem of "Reverse Nodes" however performance of the algorithm was almost same or less than the IEEE MAC in this case. Overall the algorithm achieved slightly higher throughput than 802.11 and in one scenario even achieved almost 200% increase.

7.4 Future area of investigation

There are significant opportunities for future work. Since the PHY layer parameters play a major role in performance of 802.11, getting down "good" values for these parameters in an important task. Analyzing the behavior of 802.11 and coming out with an optimum PHY layer parameter values would be quite useful.

Another aspect which this work does not analyze is the "Reverse Node". This work has been left open for future development and as an extension to this work.

Appendix A

Data Structures Used

A.1 Specific to this proposed solution

The message received is maintained at all the nodes. No duplicate ACKs and DATA packets are stored. This is required to gather the information of the on going transmissions. The state of the Node is stored as an enum defined as below.

```
/*
 *The Node Msg Recvd Structure.
 *This strucutre will include all the information related to a particular
 *messages recvd by the Node. Based on the sequence of
 *of msgs the decision will be taken whether the node is an Exposed Node
*/
     struct glomo_msgRecvd_str
     {
       NODE_ADDR sourceAddr;
       NODE_ADDR destAddr;
       unsigned short frameType;
       GlomoMsgRecvd *nextMsg;
       long int duration;
     };

     typedef enum
     {
       M802_11_EXPOSED = 0,        // 0
       M802_11_BUSY = 1,        // 1
       M802_11_WAIT = 2               // 2
     }
M802_11_MacMsgRecvd;
```

A.2 Node

The Node is represented as the following structure. It maintains simulation related Data, such as Mobility information, node position, number of nodes etc. This work required the last few fields also to be stored at each node. It maintains a list of Nodes that the current node is exposed to, sequence of RTS/CTS messages received. This makes up the information about the neighborhood that the node has to store.

```
/* Common Information  about each node. */

/* This field represents the simulation id of the node. It is used
only for simulation purposes and should not be used by the protocol
code at any layer. For the network address of the node use the next
  field, which is called nodeAddr. */

unsigned id;
NODE_ADDR nodeAddr;                  // Network address of Node
unsigned short seed3;               // Random number generator Seed
unsigned short initialSeedValue3;  // First seed value
long numNodes;                       // Number of Nodes
GlomoCoordinates position;
SplayTree splayTree;
GlomoPartition *partitionData;
GlomoMobility mobilityData;
   /* Information about partition nodes */
GlomoNode *prevNodeData;
GlomoNode *nextNodeData;
   /* Layer specific information about each node. */
GlomoProp *propData;                    // propagation information
GlomoRadio *radioDataMAX_NUM_RADIOS;  // radio layer information
int numberRadios;
GlomoMac *macDataMAX_NUM_INTERFACES;  // mac layer information
int numberInterfaces;
GlomoNetwork networkData;             // network layer information
GlomoTransport transportData;         // transport layer information
GlomoApp appData;                     // application layer information
int eotCalculatorBackPtrIndex;        // For Parallel Lookahead Calculation
   //Specific to Exposed Node solution
M802_11_MacMsgRecvd nodeStatus;       // Status : Exposed or occupied
GlomoMsgRecvd *msgRecvd;
short int *isExposedTo;
short int isReset;
```

A.3 MAC Layer

This stores the information for the MAC at each node. It manages the NAV, status of MAC, category wise simulation statistics and other simulation required parameters.

```
/* 802.11 data structure. */
typedef struct glomo_mac_802_11_str
{
    GlomoMac* myGlomoMac;
   int state;       // Mac states.
   int prevState;
   BOOL IsInExtendedIfsMode;
   clocktype noResponseTimeoutDuration;
   clocktype CW;
   clocktype BO;    // Backoff value at this station.
   clocktype lastBOTimeStamp;
   unsigned int timerSequenceNumber;
   SeqNoEntry *seqNoHead;
   int currentFrag;
   clocktype NAV;
   NODE_ADDR waitingForAckOrCtsFromAddress;
   long bandwidth;
   clocktype extraPropDelay;
   clocktype ctsOrAckTransmissionDuration;
   /* Statistics collection variables. */
   long pktsToSend,    pktsLostOverflow,
     pktsSentUnicast,  pktsSentBroadcast,
     pktsGotUnicast,   pktsGotBroadcast,
     retxDueToCts,     retxDueToAck,
     retxDueToFragAck, pktsDropped,
     fragsDropped;
   int rtsPacketsIgnoredDueToBusyChannel, rtsPacketsIgnoredDueToNAV;
   NetworkQueueingPriorityType currentPriority;
 /* Specific to Exposed Node Solution */
    clocktype rtsWaitDuration;
    int maxExposedPacketSize;
} GlomoMac802_11;
```

Various states of MAC as recorded in Glomosim are shown below.

```
/* MAC States */
   M802_11_S_IDLE,                 // 0
   M802_11_S_WFNAV,                // 1
   M802_11_S_WF_DIFS_OR_EIFS,      // 2
   M802_11_S_BO,                   // 3
   M802_11_S_NAV_RTS_CHECK_MODE,   // 4
 // Waiting For Response States
   M802_11_S_WFCTS,            // 5 First State in range
   M802_11_S_WFDATA,           // 6
   M802_11_S_WFACK,            // 7
   M802_11_S_WFFRAGACK,        // 8 Last State in range
 // Transmission States:
   M802_11_X_RTS,              //  9  First State in range
   M802_11_X_CTS,              // 10
   M802_11_X_UNICAST,          // 11
   M802_11_X_BROADCAST,        // 12
   M802_11_X_FRAGMENT,         // 13
   M802_11_X_ACK,              // 14
   M802_11_X_FRAGACK           // 15  Last State in range
```

A.4 Control Packets

The default RTS / CTS packets were changed by adding a new field, “Source Address”.

```
/*
* Standard CTS and ACK frames.
* Note: All frames types must match the short control (this one)
*       exactly for its first four (universal) fields.
*/

typedef struct _Mac802_11SCtrlFrame
{
    unsigned short frameType;
    char Padding2;
    int duration;
    NODE_ADDR destAddr;
} M802_11ShortControlFrame;
```

```
/* Modified CTS/ACK frames. Standard RTS frame is the same as below.*/
typedef struct _Mac802_11LCtrlFrame
{
   unsigned short frameType;
   char Padding2;
   int duration;
   NODE_ADDR destAddr;
   NODE_ADDR sourceAddr;
   char FCS4;
} M802_11LongControlFrame;
```

A.5 Methods specific to this solution

The following methods were added to Glomosim for implementation. The names are self explanatory. All the methods return the current status of the Node after the desired operation.

```
M802_11_MacMsgRecvd checkExposedNode(
    GlomoNode * node,
    GlomoMac802_11 * M802,
    const Message * msg);

void Mac802_11ForceTransmitFrame (
    GlomoNode * node,
    GlomoMac802_11 * M802,
    int packetSize,clocktype delay);

M802_11_MacMsgRecvd resetExposedNodeStatus(
    GlomoNode * node,
    GlomoMac802_11 * M802);
```

Bibliography

[1] L. Kleinrock and F.A. Tobagi. *Packet switching in radio channels: Part I carrier sense multiple-access modes and their throughputdelay characteristics.* IEEE Trans. Commun. COM-23(12) 1400 1416, 1975.

[2] L. Kleinrock and F.A. Tobagi. *Packet switching in radio channels: Part II the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution,.* IEEE Trans. Commun. COM-23(12) (1975) 1417, 1975.

[3] IEEE Computer Society. *802.11 Wireless Local Area Network.* IEEE Communications Magazine, Sep '99.

[4] P. Karn. *MACA - a new channel access method for packet radio.* AARUCRRL Amateur Radio 9th Computer Networking Conference, pages 13440., 1990.

[5] C. L. Fullmer and J. J. Garcia-Luna-Aceves. *Solutions to hidden terminal problems in wireless networks.* Proceedings of ACM Sigcomm 97,, Sept 97.

[6] Vaduvur Bharghavan and Alan Demers et al. *MACAW: A media access protocol for wireless LAN s.* SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications, pages 212 225,, August 1994.

[7] M. Gerla, K. Tang, and R. Bagrodia. *TCP Performance in Wireless Multi-hop networks.* IEEE WMCSA 99, Feb 99.

[8] Rajiv Bagrodia and Mario Gerla et al. *Glomosim: A Scalable Network Simulation Environment.* UCLA, 2000.

[9] Theodore S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, New Jersey, 1996.

[10] Xiaoyan Hong and Mario Gerla et al. *A Group Mobility Model for Ad Hoc Wireless Networks.*

[11] S. XU and T. Saadawi. *Does IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?,*. IEEE Communications Magazine,, June, 2001.

[12] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *Mobile Computing and Networking*, pages 61–69, 2001.

[13] R. Bagrodia and R. Meyerr et al. *PARSEC: A Parallel Simulation Environment for Complex System ,*. UCLA technical report,, 1997.